3-2021

Structurally enriched entity mention embedding from semi-structured textual content

Lee Hsun HSIEH
Singapore Management University, lhhsieh@smu.edu.sg

Yang Yin LEE
Singapore Management University, yylee@smu.edu.sg

Ee-Peng LIM
Singapore Management University, eplim@smu.edu.sg

Structurally Enriched Entity Mention Embedding from Semi-structured Textual Content

Lee-Hsun, Hsieh
Singapore Management University
Singapore
jesus255221@gmail.com

Yang-Yin, Lee
Singapore Management University
Singapore
yylee@smu.edu.sg

Ee-Peng, Lim
Singapore Management University
Singapore
eplim@smu.edu.sg

ABSTRACT

In this research, we propose a novel and effective entity mention embedding framework that learns from semi-structured text corpus with annotated entity mentions without the aid of well-constructed knowledge graph or external semantic information other than the corpus itself. Based on the co-occurrence of words and entity mentions, we enrich the co-occurrence matrix with entity-entity, entity-word, and word-entity relationships as well as the simple structures within the documents. Experimentally, we show that our proposed entity mention embedding benefits from the structural information in link prediction task measured by mean reciprocal rank (MRR) and mean precision@K (MP@K) on two datasets for Named-entity recognition (NER).

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; **Information extraction**; *Ontology engineering*;

KEYWORDS

Entity mention embedding, structural enrichment

ACM Reference Format:
Lee-Hsun, Hsieh, Yang-Yin, Lee, and Ee-Peng, Lim. 2021. Structurally Enriched Entity Mention Embedding from Semi-structured Textual Content. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3412841.3442100

1 INTRODUCTION

Recent years have witnessed the increasing popularity of contextualized embeddings [1, 6]. There has been very little research on learning word and entity embeddings from semi-structured text corpuses which are not associated with rich semantic knowledge often found in well-structured knowledge bases. A straightforward way of learning the embeddings of these entities and words is to directly apply word embedding models (e.g., GloVe [4]) on the document treating both entities and words as words. This approach however ignores the document structure which can play a vital role in determining whether two terms should share similar semantics. To address this issue, we offer an alternative idea and propose a lightweight framework to learn entity[1] embeddings from datasets where the knowledge graph of entities is not available but structures exist in documents.

Description
Knowledge of Anti Money Laundering transaction monitoring red flags ...
Strong data analysis skills ...
Very good understanding of KYC, AML Transaction monitoring ...
Requirement
Oracle Mantas FS data model and scenarios ...
Basic sql skills
Working knowledge of Oracle applications such as EDQ / DIH

Figure 1: An example in Jobs dataset. Underlines in the figure indicate skill entities annotated by annotators.

Specifically, we propose a framework that learns entity embeddings using the document structure found in the semi-structured text containing the entities. Consider the example document in Figure 1 which is an annotated job post with marked-up *skill* entities. This document contains three levels of structure, i.e., *sentence*, *section* ("Description" and "Requirement"), and *document* levels. Intuitively, the skills *EDQ* and *DIH* in the same sentence should be semantically closer to each other than *EDQ* and another skill *sql* from different sentences of the same section. The *EDQ* and another skill *sql* entity pair however is in turn closer than the entity pair *sql* and *data analysis* which exist in the same document but different sections. While there is no existing knowledge graph covering these entities, we can use the structural information of the documents to enhance the relationship between entities so as to learn their embeddings well. Our proposed enrichment procedure can give additional small weights to a pair of entities that do not co-occur within a fixed size window but are still related by document structure. We call this *structure enrichment*. Besides enriching entity-entity pairs, we also consider the entity-word and word-entity pairs. When enriching an entity-word pairs, the entities in the document are treated as target terms and the words are treated as context terms. This structure enrichment approach can be performed on an co-occurrence matrix of entities and words, which cover the entity-entity, entity-word and word-entity pairs. Note that our proposed enriching procedure is a general framework and can be applied to different semi-structured NER datasets.

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8104-8/21/03.
https://doi.org/10.1145/3412841.3442100

[1] *Entity mention* should be a more precise term as this work focuses on learning the entity mention embedding (without canonicalization). For simplicity, we use *entity* to mean entity mention throughout this paper unless otherwise specified. Similarly, we use *word* to represent a *normal word* (non-entity word).

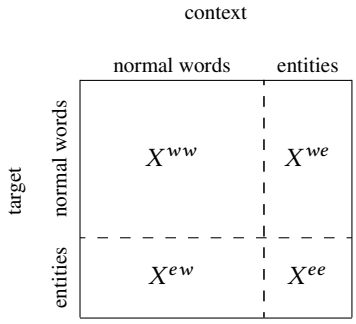


Figure 2: The co-occurrence matrix X.

In our problem formulation, we treat the vocabulary of terms (entities and words) as nodes in a graph and the the co-occurrence count of a pair of terms as the edge weight between two terms. We apply three popular network embedding models DeepWalk [5], node2vec [2], and LINE [7] and a word embedding model GloVe [4] on link prediction task. We find that our proposed structural enriched entity embedding model outperforms the embeddings learned without structural information using CoNLL-2003 [8] and Jobs datasets.

2 METHODOLOGY

Given a dataset I with $|I|$ annotated documents, let $G = (V, E)$ be a graph with nodes representing a vocabulary of two types of terms, $V = V_w \cup V_e$. V_e refers to the set of all the entities[2] and V_w refers to the set of all the words in I. All these entities and words are at surface form. $E \subseteq (V \times V)$ denotes a set of edges. The weight of an edge $(v_i, v_j) \in E$ represents how strong the two terms are related. A natural way to represent these weights is to construct a co-occurrence matrix similar to that in GloVe [4]. Specifically, the entry X_{ij} tabulates the occurrence count of term j in the context of target term i with distance weight decay. As the vocabulary contains V_w and V_e, X consists of four parts: X^{ww} (word–word), X^{we} (word–entity), X^{ew} (entity–word), and X^{ee} (entity–entity) as shown in Figure 2.

2.1 Enriching Entity-Entity Co-occurrences

Next, we leverage on the structure of document to enrich the co-occurrence between every pair of terms. The structural enriching procedure updates X^{ee} by the context. To meet our objective, the co-occurrence counts are computed on the same dataset. Given a document I_d (e.g., a job post) in I with n entity instances $\{\hat{e}_1, \cdots, \hat{e}_n\}$, each pair of instances $(\hat{e}_p, \hat{e}_q) \in \{\hat{e}_1, \cdots, \hat{e}_n\}$ for $p \neq q$ appears in the same structural element at certain level, e.g., document, section, sentence, etc. Suppose there are L predefined structure levels, we define $scsl(\hat{e}_p, \hat{e}_q)$ as the **smallest common structure level** of entity instances $\hat{e}_p$ and $\hat{e}_q$. We define level 1 to be the finest structure and level L to be the coarsest structure. For example, for a dataset with sentences found in documents, we assign levels 1 and 2 to sentence and document structural elements, $(sentence(1), document(2))$, respectively. If $\hat{e}_p$ and $\hat{e}_q$ belong to two distinctive sentences but in the same document, $scsl(\hat{e}_p, \hat{e}_q) = 2$ as their smallest common structure level is $document$. Suppose e_p and e_q are the surface forms of $\hat{e}_p$ and $\hat{e}_q$ respectively, our proposed procedure updates X_{e_p,e_q} for every pair of entity instances $(\hat{e}_p, \hat{e}_q) \in \{\hat{e}_1, \cdots, \hat{e}_n\}^2$ for $p \neq q$ with some additional weight as follows:

$$X_{e_p,e_q} \leftarrow X_{e_p,e_q} + f(\hat{e}_p, \hat{e}_q) \tag{1}$$

where $f(\hat{e}_p, \hat{e}_q) = \alpha_l$ if $scsl(\hat{e}_p, \hat{e}_q) = l$. The function $f(\hat{e}_p, \hat{e}_q)$ returns the additional co-occurrence weight based on the granularity of the common structure covering both $\hat{e}_p$ and $\hat{e}_q$. As increasing level numbers correspond to fine-to-coarse levels, we expect α_l's to follow $\alpha_1 > \alpha_2 > \cdots > \alpha_L$. The update procedure is performed on all the documents in I to enrich X into the $|V| \times |V|$ matrix X'.

2.2 Enriching Entity-Word and Word-Entity Co-occurrences

In addition to the entity-entity relationship, words surrounding entities and entities surrounding words can be used to construct and enrich their co-existence expressed in the form of co-occurrence matrix. We apply a similar strategy to enrich the X^{ew} and X^{we}. Other than entity instances in a document I_d, $\{\hat{e}_1, \cdots, \hat{e}_n\}$, suppose m word instances $\{\hat{w}_1, \cdots, \hat{w}_m\}$ also exist in I_d. Similar to entity-entity enrichment, for each pair of entity instances and word instances $(\hat{e}_p, \hat{w}_q)$ where $\hat{e}_p \in \{\hat{e}_1, \cdots, \hat{e}_n\}$ and $\hat{w}_q \in \{\hat{w}_1, \cdots, \hat{w}_m\}$, we redefine $scsl$ to consider the smallest common structure level shared by them: $scsl(\hat{e}_p, \hat{w}_q)$. Let w_q be the surface form of the word instance $\hat{w}_q$. To preserve the symmetry of the matrix, We update X' for every $(\hat{e}_p, \hat{w}_q)$ pair in I_d by:

$$X_{e_p,w_q} \leftarrow X_{e_p,w_q} + f(\hat{e}_p, \hat{w}_q) \tag{2}$$

$$X_{w_p,e_q} \leftarrow X_{w_p,e_q} + f(\hat{w}_p, \hat{e}_q) \tag{3}$$

where $f(\hat{e}_p, \hat{w}_q) = \alpha_l$ if $scsl(\hat{e}_p, \hat{w}_q) = l$. The above update procedure is performed on all documents in I to enrich X' into the $|V| \times |V|$ matrix X''. Given the enriched matrices X' and X'', we also extract a small co-occurrence submatrix from the X^{ee} part of X' and X'' and call it $X^{\dagger}$. This extracted matrix $X^{\dagger}$ will largely reduce the amount of information but maintaining the structural information between the entities.

2.3 Learning Entity Mention Embeddings

After the enrichment process, we can directly apply word embedding model GloVe on matrices X, X', X'', $X^{\dagger}$. To apply network embeddings on them, we derive weighted graphs from these matrices. Each of $\{X, X', X''\}$ is turned into a graph with vertices V and edges with weights corresponding to co-occurrence values. For $X^{\dagger}$, they too are represented as weighted graphs with vertices V_e. For each of the above graphs, network embedding models such as DeepWalk, node2vec, and LINE can be applied to learn the entity embeddings.

3 EXPERIMENT

We used two datasets in our experiments, namely:

CoNLL-2003 [8]: We select the English version which contains 1,393 news articles labeled with four named entity types: persons, locations, organizations and names of miscellaneous.

Jobs: This is a small job post dataset we collected from Singapore's Jobsbank [3]. The dataset contains selected 300 job posts for each of

[2]Noted that we glue the entities and their types with a reserved underscore character to distinguish entities (e.g., *united_states_loc*) and words in the context.

[3]https://www.mycareersfuture.sg/

dataset	CoNLL-2003	Jobs
#Documents	1,393	1,800
#Sentences	22,137	33,325
#Tokens	301,418	426,187
$\|V_w\|$	4,217	3,876
$\|V_e\|$	1,402	412
#Vertices	5,619	4,288
X	1,008,427	1,666,005
X'	1,070,016	1,677,994
X''	1,568,976	2,086,334
$X^\dagger$	69,529	21,416

Table 1: Statistics of the CoNLL-2003 and Jobs datasets.

the six occupations, namely: software developer (SD), business consultant (BC), sales and marketing manager (SM), personnel/human resource officer (HR), ledger and accounts clerk (LA), and financial analyst (FA). Each job post contains two sections *Requirement* and *Description* (see Figure 1), and only skill entities are annotated.

The statistics of CoNLL-2003 and Jobs are shown in Table 1. The last four rows of Table 1 show the number of non-zero entries in the matrix. All the terms are lower-cased and a frequency threshold of 5 is set to filter out low frequency entities and words. CoNLL-2003 contains two structure levels $\{sentence(1), document(2)\}$, and we set α_1 and α_2 to 1.0 and 0.5 respectively. Jobs contains three structure levels $\{sentence(1), section(2), document(3)\}$ and we set α_1, α_2, and α_3 to 1.0, 0.5, and 0.25 respectively.

3.1 Experiment Settings

As there are no other baseline methods to be compared with, the main performance comparison is on different combinations of embedding models and enrichment methods. We select link prediction task for evaluating the enrichment methods, an evaluation task commonly used in representation learning [3, 10, 11]. In link prediction, we randomly mask out 10% of the entity-entity co-occurrences with co-occurrence values greater than the median. This is to mask out non-trivial links or co-occurrence entities.

For GloVe, the number of update iterations is set to 128 and the context window size is 15. For DeepWalk and node2vec, the number of random walks is 100 and the length of random walk is 40; both the return parameter p and in-out parameter q of node2vec are set to 1. For LINE, the number of negative samples is set to 1 and the weight of negative samples is set to 5. The dimension d_v of all the embedding models are set to 128. For link prediction task involving a masked link between e_i and e_j, we use e_i as the reference and rank the remaining entities by cosine similarity. We then use MRR and MP@K to measure the prediction accuracy. In general, higher MRR and MP@K indicate a better learnt embeddings that capture the semantics relatedness of entities.

3.2 Experiment Results

Table 2 shows the results on CoNLL-2003 and Jobs. Results with boldface show the best performing method. Among the enrichment methods, we find the ones using X'' significantly outperform those using other enrichment methods across all embedding models for the two datasets. This suggests that document structure information contributes well to link prediction in entity graph. The differences between node embedding models are rather small compared to the performance difference due to enrichment. In CoNLL-2003, the performance of $X^\dagger$ outperforms X with large margin in all the models. The execution time of $X^\dagger$ is about 8 times faster than X. Both improvements in execution time and performance suggest the potential industry applications of $X^\dagger$.

	CoNLL-2003			Jobs		
	MRR	MP@5	MP@10	MRR	MP@5	MP@10
GloVe X	3.16	3.51	6.05	13.98	19.32	26.11
GloVe X'	17.22	25.64	32.88	9.73	12.03	17.15
GloVe X''	**27.55**	**40.09**	**49.99**	**19.34**	**28.01**	**39.02**
GloVe $X^\dagger$	21.10	30.78	39.95	6.81	8.74	14.36
DeepWalk X	23.52	33.25	40.66	24.40	35.89	46.83
DeepWalk X'	38.01	58.03	71.17	18.76	26.80	38.36
DeepWalk X''	**41.19**	**61.38**	**73.73**	**25.27**	**36.46**	**50.16**
DeepWalk $X^\dagger$	37.00	56.27	70.03	21.38	30.93	43.01
node2vec X	23.35	33.47	40.63	24.39	35.77	46.23
node2vec X'	37.98	58.01	71.05	18.64	26.76	38.27
node2vec X''	**41.17**	**61.51**	**73.69**	**25.33**	**36.57**	**50.41**
node2vec $X^\dagger$	36.88	56.21	69.85	21.32	30.71	42.79
LINE X	23.63	33.10	40.89	24.37	35.74	46.30
LINE X'	38.03	58.01	71.04	18.72	26.62	38.26
LINE X''	**41.20**	**61.50**	**73.74**	**25.30**	**36.64**	**50.07**
LINE $X^\dagger$	36.91	56.27	70.04	21.48	30.57	43.52

Table 2: Link prediction results on CoNLL-2003 and Jobs.

3.3 Visualization

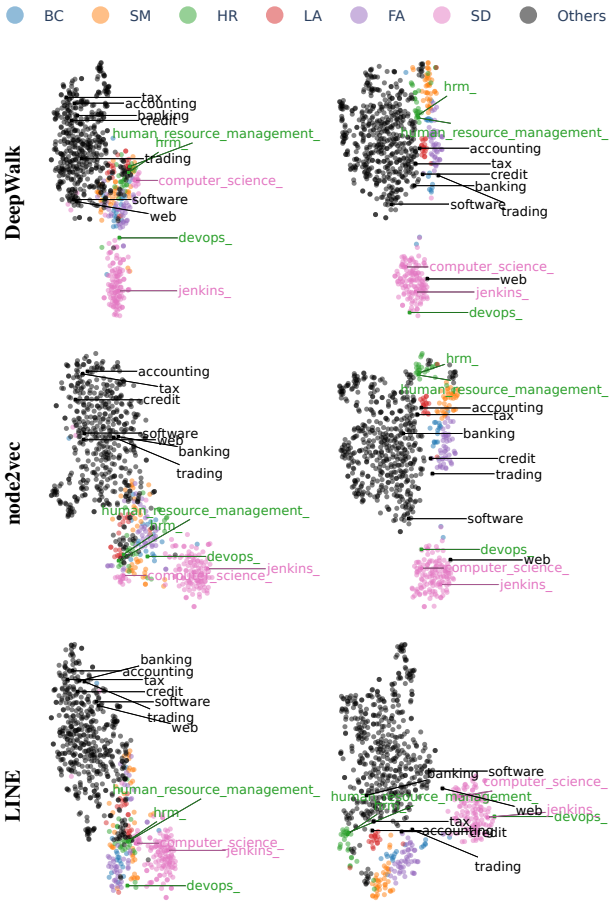
Figure 3 shows the visualization using t-SNE [9]. Each node with color other than black represents a skill in V_e of some occupation. We use 50% as a threshold to determine which occupation the skill belongs to (i.e., an entity e belongs to *BC* if over 50% of the time e is in *BC*). "Others" refers to the top 500 words in V_w by frequency (e.g., *experience*). From Figure 3, we find that all node embedding models with X'' separate different occupations' skills and words (*Others*) better than that using X. For the latter, only the entities in *SD* are well separated from entities of other occupation labels. In particular, we find *hrm_* and *human_resource_ management_*, which are aliases of each other, are almost on the same point in X'' but somehow separated in X.

Also, by observing the normal words (*Others*) shown in the figures, models with X'' have the ability to bring related normal words closer to their corresponding occupation. For example, *web* and *software* are away from the SD skill cluster in X but close to SD skill cluster in X''.

Another example is the *devops_* skill in HR. *devops_* has 58% of its occurrences in HR job posts and 25% in SD job posts. We find that some job posts in HR contain *devops_*, *computer_science_*, and *jenkins_* at the same time. We suspect that those job posts have been wrongly classified with HR label. They should be better assigned with SD label. Nevertheless, network embedding models with X'' successfully learn that *devops_* should be closer to SD skills than HR skills. This may due to the fact X'' considers other related terms in the same job post (e.g., *computer_science_* and *jenkins_*).

3.4 Case Study

We select the ORG entity "ducati_org" from CoNLL-2003 as a case example to show the different results between LINE X'' and LINE X as shown in Table 3. For LINE X, the top 4 entities include one



(a) X **(b)** X''

Figure 3: Visualization of the embedding results.

target	retrieved	X''	X
ducati_org	ford_escort_misc	9	1
	isolde_kostner_per	38	2
	florence_masnada_per	39	3
	katja_seizinger_per	42	4
	yoshikawa_per	1	73
	slight_per	2	47
	kocinski_per	3	38
	aoki_per	4	70

Table 3: ducati_org and the top 4 ranked entities by LINE X' and LINE X.

car name and three Olympic athletes' names while the top 4 entities for LINE X'' are MotoGP riders. We suspect the reason is because CoNLL-2003 contains racing result articles as shown in Figure 4. For LINE X, the default context window size of 15 fails to capture entities from neighboring sentences in racing result articles such as the "kocinski_per" and "slight_per" because the newline symbols separate them into two different sentences. On the other hand, the structural enriched LINE X'' successfully captures this information and strengthens the co-occurrence between these entities and the target.

fastest lap : aoki_per 147.786 kph
world championship standings (after nine rounds) :
1. slight_per 280 points
2. corser_per 269
3. kocinski_per 254
4. fogarty_per 236
5. colin_edwards_per (u.s._loc) yamaha_org 176
6. pier_francesco_chili_per (italy_loc) **ducati_org** 175

Figure 4: A snapshot of CoNLL-2003.

4 CONCLUSION AND FUTURE WORKS

This paper proposes a novel and effective entity mention embedding learning framework that learns from semi-structured NER datasets. Starting from the window-based co-occurrence counts of the word and entity terms in the documents, we further enrich the co-occurrence matrix with entity-entity, entity-word and word-entity relationships derived from document structure. Experimentally, we show that the structurally enriched co-occurrence matrix can contribute to learning of entity embeddings that capture the semantic relatedness among entities for more accurate link prediction accuracy.

In the future, we plan to explore contextualized embedding models on semi-structured datasets to compare with our proposed framework. How to integrate the canonicalization method and linking to knowledge bases into the semi-structured data is another direction of research.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.
[2] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *KDD*.
[3] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, Elena Baralis, Michele Osella, and Enrico Ferro. 2018. Knowledge graph embeddings with node2vec for item recommendation. In *ESWC*. Springer, 117–120.
[4] Jeffrey Pennington, Richard Socher, and Christopher D Manning. [n. d.]. Glove: Global vectors for word representation. In *EMNLP*.
[5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *KDD*.
[6] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL*.
[7] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-Scale Information Network Embedding. In *WWW*.
[8] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *HLT-NAACL*.
[9] Laurens van der Maaten. 2014. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15, 93 (2014), 3221–3245.
[10] Hao Wu, Hanyu Zhang, Peng He, Cheng Zeng, and Yan Zhang. 2019. A Hybrid Approach to Service Recommendation Based on Network Representation Learning. *IEEE Access* 7 (2019), 60242–60254.
[11] Denghui Zhang, Junming Liu, Hengshu Zhu, Yanchi Liu, Lichen Wang, Pengyang Wang, and Hui Xiong. 2019. Job2Vec: Job title benchmarking with collective multi-view representation learning. In *CIKM*. 2763–2771.